Using Model Checking to Validate AI Planner Domain Models

John Penix, Charles Pecheur

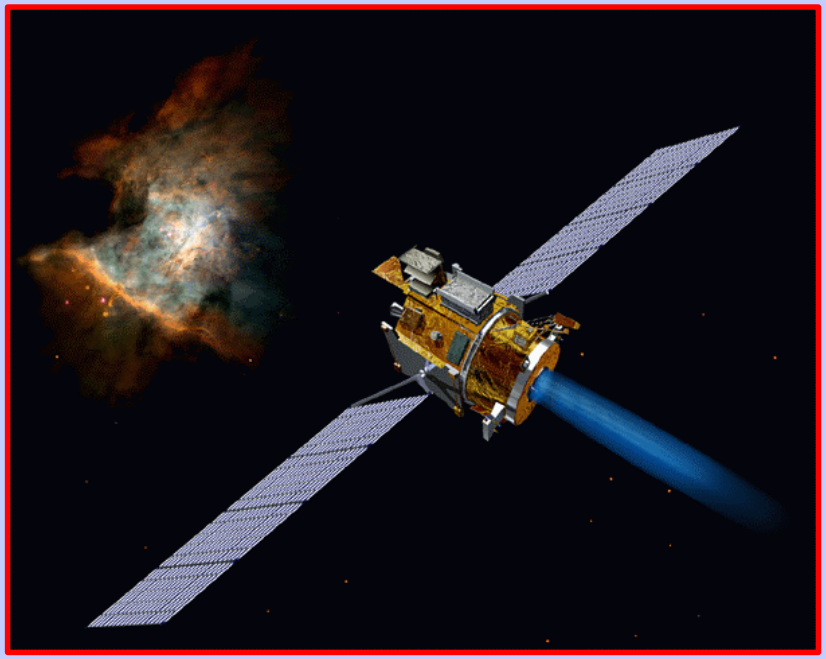
and Klaus Havelund

Automated Software Engineering

NASA Ames Research Center

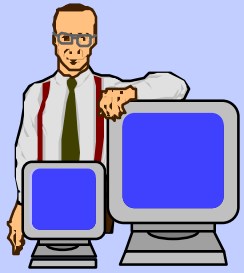
The Problem: High-Assurance Autonomous Systems

- **Long lifetimes**
- **Limited Access**
- **Reactive/Adaptive**
- **Agent-Based Architectures**
- **Advanced Algorithms**
- **Knowledge-Based**



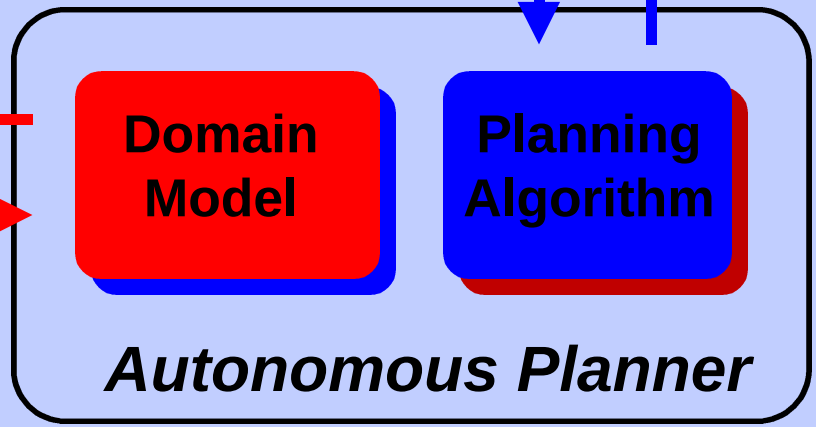
Verification and Validation of Autonomous Planning Systems

- *Our approach is to separate algorithm verification from domain model validation*



Computer Scientists

Domain Engineers

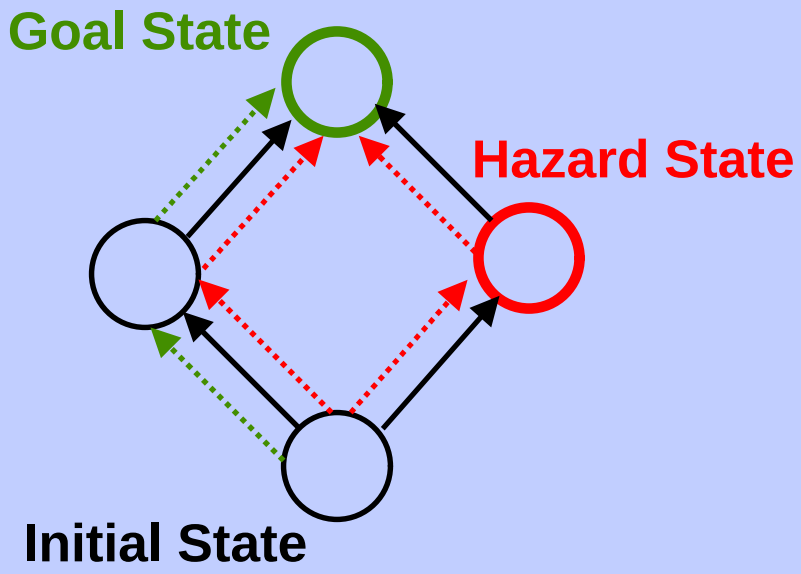


Model Checking

- *Technique for verification of finite-state systems - traditionally used for hardware*
- *Determines whether a FSM is a "model" of a temporal logic formula*
- *Exhaustive - evaluates **all** possible executions of events in the system*
- *Allows non-deterministic modeling/abstraction of the "environment"*
- *Limited by "state space explosion"*

Model Checking vs. Planning

- ***Exhaustive*** *model checking algorithms can find flaws in domain models that may not be discovered by testing with a* ***heuristic*** *planner*



The planner may find a path to the goal without discovering the hazard state. Model checking tries all paths and finds the hazard state.

The HSTS Planner's Domain Description Language

- *Object-oriented data structures with qualitative and quantitative constraints on variable values*

Robot
:state variables
At: {A,B,C}
Task: {Move,Fix,Rest}
Charge: {Empty,Full}

Hole
:state variables
At: {A,B,C}
Status: {Exists,Fixed}

((Robot.Task=Fix) starts_before (10 20) (Hole.Status = Fixed))

Model Checking for DDL

- *We developed a translation from a DDL model to a finite state transition model*
- *Tested the translation on 3 model checkers:*
  - *Spin (from Bell Labs)*
  - *Murphi (from Stanford University)*
  - *SMV (from Carnegie Mellon University)*

Results: Expressibility

- *General translation to finite state transition model able to express qualitative constraints with temporal locality*
- *No support for quantitative constraints*
- *Temporally distant relationships require special consideration:*
  - *history variables in Spin & Murphi*
  - *fairness constraints in SMV*

Results: Performance

- *Experimented on "small" model of autonomous robot with 65 temporal constraints*
- *The model had 16320 reachable states out of 559872 potential states (highly constrained)*
- *Exhaustive verification with Spin & Murphi in under 30 seconds*
- *Exhaustive verification with SMV in 0.05 seconds (due to better target language)*

Results: Capabilities

- *Model checking "error traces" are plans*
- *Able to perform analysis that are not directly supported by testing with the planner:*
  - *Is there a plan (a path from an initial state to a goal state) for **any** legal initial state?*
  - *Is there a plan for **every** legal initial state?*
  - *Is there a plan from every reachable state?*
  - *Can I reach a state where X is true?*

Conclusions

- *Model checking has the potential to overcome limits to model validation using a planner*
- *Model checking can be used to effectively validate "simple" domain models*
- *Further experiments are necessary to see if temporally distant relations and quantitative constraints can be effectively supported*